

# Problem Set 2

*Denits Panova, Thomas Vicente, Felix Gutmann*

*15 May 2016*

## Problem 1

a)

(1)

Define the utilities as:

$$\begin{aligned} U_i(A) &:= \alpha + d_i^A a \\ U_i(B) &:= \beta + d_i^B b, \end{aligned}$$

where  $\alpha, \beta \geq 0$  and  $d_i^*$  is the number ...

Best reply:

$$U_i(A) > U_i(B)$$

Substitute the definition of the utilities in the last equation:

$$\alpha + d_i^A a > \beta + d_i^B b$$

Define  $d_i$  and rephrase it:

$$\begin{aligned} d_i &:= d_i^A + d_i^B \\ \underbrace{\frac{d_i^A}{d_i}}_{f_i} + \underbrace{\frac{d_i^B}{d_i}}_{(1-f_i)} &= 1 \end{aligned}$$

Using the last identity we can write the the inequality in the following way:

$$\begin{aligned} \alpha + f_i a &> \beta + (1 - f_i) b \\ f_i(a + b) &\geq \beta + b - \alpha \\ f_i &\geq \frac{(\beta - \alpha) + b}{b + a} \in [0, 1] \end{aligned}$$

Let  $p_A = \frac{(\beta - \alpha) + b}{b + a}$  and  $p_B = (1 - p_A) = \frac{(\alpha - \beta) + a}{b + a}$ . So best response is to choose A **iff** at least  $p_A$  of its friends choose A.

(b)

The Nash-Equilibrium (NE) appears when:

- any node  $i$  that chooses A has at least  $p_A$  friends who also choose A
- any node  $i$  that chooses B has at least  $p_B$  friend who also choose B

(c)

Let  $C_p$  a  $p$ -cluster and  $C_p \subset G$ . In the  $p$ -clustering sense we can say that NE occurs if  $i$  chooses A then  $i \in C_A$ , where  $C_A$  is a  $p_A$ -cluster and if  $i$  chooses B then  $i \in C_B$ , where  $C_B$  is a  $p_B$ -cluster.

b)

Set  $\alpha = \beta = 0$ . Each nodes starts with B. Let  $t = q = \frac{1}{2}$  for switching to A.

(b1)

Let  $S = \{e - f\}$ , where S is the starting set meaning that the two nodes initially adopt behaviour A.

!!!! FOR FELIX !!!!! put in a graph or leave some space for drawing one #Denisisthebest

(b2)

The Blocking set is  $BS := \{g, j\}$ , with density  $D = \frac{2}{3}$ .

c)

!!! DENI WILL PAINT SOME MORE STUFF!!!

d)

No we can't have the same Nash-Equilibria, because when we have a tie we are more likely to choose one of them namely to choose the one which has bigger initial values ( $\alpha$  or  $\beta$ ).

## Problem 2

(2.1.a)

Step 1:  $h_{A_i}^{(1)} = 2h_{C_i}^{(1)} = \frac{1}{8} a_{B_i}^{(1)} = \frac{3}{8} a_D^{(1)} = \frac{5}{8}$  Step 2:  $h_{A_i}^{(2)} = \frac{9}{8} a_{B_i}^{(2)} = \frac{27}{8} h_{C_i}^{(2)} = \frac{5}{8} a_D^{(2)} = \frac{25}{8}$  Conclusion:  $a_{B_i}^{(2)} > a_D^{(2)}$  even from the second iteration there is a difference between the authority measures of the two different nodes. (2.1.b)

We define the formulas for the two sub-graphs separatly. The first sub-graph consists of A's and B's. Let  $b :=$  number of B's and  $a :=$  number of A's and  $c =$  constant  $= \frac{1}{8}$ . For Step  $k = 1$ :  $h^{(1)} = a^0 b^0 c = c$   $a^{(1)} = ab^0 c$  Step  $k = 2$ :  $h^{(2)} = a^{(1)} b = bab^0 c = b^{(k-1)} a^{(k-1)} c$   $a^{(2)} = abab^0 c = a^{(k)} b^{(k-1)} c$  Step  $k = 3$ :  $h^{(3)} = a^{(2)} b = a^{(k-1)} b^{(k-2)} bc = a^{(k-1)} b^{(k-1)} bc$   $a^{(3)} = aa^{(k-1)} b^{(k-1)} c = a^{(k)} b^{(k-1)} c$  In general we define:  $h_{A_i}^{(k)} := a^{(k-1)} b^{(k-1)} bc$   $a_{B_i}^{(k)} := a^{(k)} b^{(k-1)} c$

for the secon sub-graph Step  $k = 1$ :  $h^{(1)} = c$   $a^{(1)} = 5c$  Step  $k = 2$ :  $h^{(2)} = 5c = 5^{(k-1)} c$   $a^{(2)} = 5 * 5c = 5^k c$  In general we define:  $h_{C_i}^{(k)} = 5^{(k-1)} c$   $a_D^{(k)} = 5^k c$

(2.1.c)

Let the normalizing constant be

$$\begin{aligned} const_h &= 3[a^{k-1} b^{k-1} c] + 55^{k-1} c \\ &= 33^{2k-2} c + 5^k c \\ &= (3^{2k-1} + 5^k) c \end{aligned}$$

The normalized version of the hub measure is:

$$\begin{aligned} \widehat{h_{A_i}} &= \frac{h_{A_i}}{const_h} \\ &\propto \frac{9^k}{(3^{2k-1} + 5^k)} \widehat{h_{C_i}} = \frac{5^k}{3^{2k-1} k + 5^k} \end{aligned}$$

The other authority normalizing constant is defined as:

$$\begin{aligned} const_a &= 3a^K b^{k-1}c + 5^k c \\ &= 3^k c + 5^k c \\ \widehat{a_{B_i}} &= \frac{3^{2k-1}}{3^K b^{k-1} + 5^k} \\ &\propto \frac{9^K}{3^K + 5^K} \\ \widehat{a_D} &= \frac{5^k}{3^K + 5^k} \end{aligned}$$

Conclusion: As  $k$  goes to infinity  $\widehat{a_{B_i}}$  will go slower to 0 than  $\widehat{a_D}$ . The same is equivalent for  $\widehat{h_{A_i}}$  than  $\widehat{h_{C_i}}$ . So when pages have multiple reinforcement, meaning hub points to more than one page, then in the world web the authorities will be bigger and could more easily be found.

(2.2)

Then the effect will be the same for the two sub-graph, even it will be accelerated.

(2.3)

SIGNEDSEALEDDELIVERED

```
setwd('/Users/felix/Documents/GSE/Term 3/14D009 Social and Economic Networks/Problemsets/14D009-Social-
library(igraph)
```

```
##
## Attaching package: 'igraph'

## The following objects are masked from 'package:stats':
##
##   decompose, spectrum

## The following object is masked from 'package:base':
##
##   union
```

```
#1.2
g <- sample_bipartite(10, 5, p=.4)
gp = bipartite_projection(g)$proj1
H = hub_score(gp)$value
E = eigen_centrality(gp)$value
H ; E
```

```
## [1] 43.1544
```

```
## [1] 6.567211
```

```
round(H,3) == round(E,3)
```

```
## [1] FALSE
```

```
round(H,3) == round(E^2,3)
```

```
## [1] FALSE
```

```
#2.1
```

```
probas = c(.01, .1, .25, .6)
```

```
for (p in probas) {
```

```
  print(paste('For probability =', p, ':'))
```

```
  n = 5
```

```
  g = erdos.renyi.game(n, p)
```

```
  avg = average.path.length(g)
```

```
  print(paste('      - average distance =', avg))
```

```
  d = get.diameter(g)
```

```
  print(paste('      - diameter =', length(d) - 1))
```

```
  D <- lower.tri(shortest.paths(g))
```

```
  D[D==Inf] <- 0
```

```
  print(paste('      - the asbolute difference between the average distance and its approximation is', abs(avg-m
```

```
  print(paste('      - the asbolute difference between the diamter and its approximation is', abs(avg-m
```

```
  d <- degree_distribution(g)
```

```
  pdf(paste0('plot_p', p, '.pdf'))
```

```
  plot(d, type='l')
```

```
  dev.off()
```

```
  cl = clusters(g)
```

```
  print(paste('      - number of components =', cl$no))
```

```
  print('      - we see in the following table:')
```

```
  t = table(cl$size)
```

```
  print(t)
```

```
  print('')
```

```
  print(paste('      that the largest component captures', ifelse(max(as.numeric(rownames(as.matrix((t
```

```
  print('')
```

```
  print('')
```

```
}
```

```
## [1] "For probability = 0.01 :"
```

```
## [1] "      - average distance = NaN"
```

```
## [1] "      - diameter = 0"
```

```
## [1] "      - the asbolute difference between the average distance and its approximation is NaN ."
```

```
## [1] "      - the asbolute difference between the diamter and its approximation is NaN ."
```

```
## [1] "      - number of components = 5"
```

```
## [1] "      - we see in the following table:"
```

```
##
```

```
## 1
```

```
## 5
```

```
## [1] ""
```

```

## [1] "      that the largest component captures less than 80 percent of the nodes."
## [1] ""
## [1] ""
## [1] "For probability = 0.1 :"
## [1] "      - average distance = 1"
## [1] "      - diameter = 1"
## [1] "      - the asbolute difference between the average distance and its approximation is 0.5 ."
## [1] "      - the asbolute difference between the diamter and its approximation is 0 ."

## [1] "      - number of components = 4"
## [1] "      - we see in the following table:"
##
## 1 2
## 3 1
## [1] ""
## [1] "      that the largest component captures less than 80 percent of the nodes."
## [1] ""
## [1] ""
## [1] "For probability = 0.25 :"
## [1] "      - average distance = 1.33333333333333"
## [1] "      - diameter = 2"
## [1] "      - the asbolute difference between the average distance and its approximation is 0.8333333333"
## [1] "      - the asbolute difference between the diamter and its approximation is 0.33333333333333 ."

## [1] "      - number of components = 3"
## [1] "      - we see in the following table:"
##
## 1 3
## 2 1
## [1] ""
## [1] "      that the largest component captures less than 80 percent of the nodes."
## [1] ""
## [1] ""
## [1] "For probability = 0.6 :"
## [1] "      - average distance = 1.3"
## [1] "      - diameter = 2"
## [1] "      - the asbolute difference between the average distance and its approximation is 0.8 ."
## [1] "      - the asbolute difference between the diamter and its approximation is 0.3 ."

## [1] "      - number of components = 1"
## [1] "      - we see in the following table:"
##
## 5
## 1
## [1] ""
## [1] "      that the largest component captures more than 80 percent of the nodes."
## [1] ""
## [1] ""

```

## #2.2

```

d <- read.table('/Users/felix/Documents/GSE/Term 3/14D009 Social and Economic Networks/Problemsets/14D009
a <- graph.adjlist(as.matrix(d))

```

```
#centrality measures
a1 = centr_degree
a2 = eigen_centrality
a3 = page_rank
a4 = authority.score
```

```
#... their values
a1(a)$centralization
```

```
## [1] 0.0006354772
```

```
a2(a)$value
```

```
## [1] 31.82109
```

```
a3(a)$value
```

```
## [1] 1
```

```
a4(a)$value
```

```
## [1] 1008
```

```
#... and the 10 most central instances according to each of them
```

```
c1 = a1(a)$res
c2 = a2(a)$vector
c3 = a3(a)$vector
c4 = a4(a)$vector
maxC <- function(v) {unlist(as.matrix(sort.int(v, decreasing=T, index.return=TRUE))[2]))[1:10]}
print('degree centrality')
```

```
## [1] "degree centrality"
```

```
maxC(c1)
```

```
## [1] 53213 35290 38109 62821 93504 92790 21718 1086 89732 111161
```

```
print('eigen centrality')
```

```
## [1] "eigen centrality"
```

```
maxC(c2)
```

```
## [1] 53213 101361 101495 101585 101446 101262 101520 101370 101558 101610
```

```
print('page rank')
```

```
## [1] "page rank"
```

```
maxC(c3)
```

```
## [1] 105680 81761 54526 92790 50163 22653 23289 39808 92043 36907
```

```
print('HITS')
```

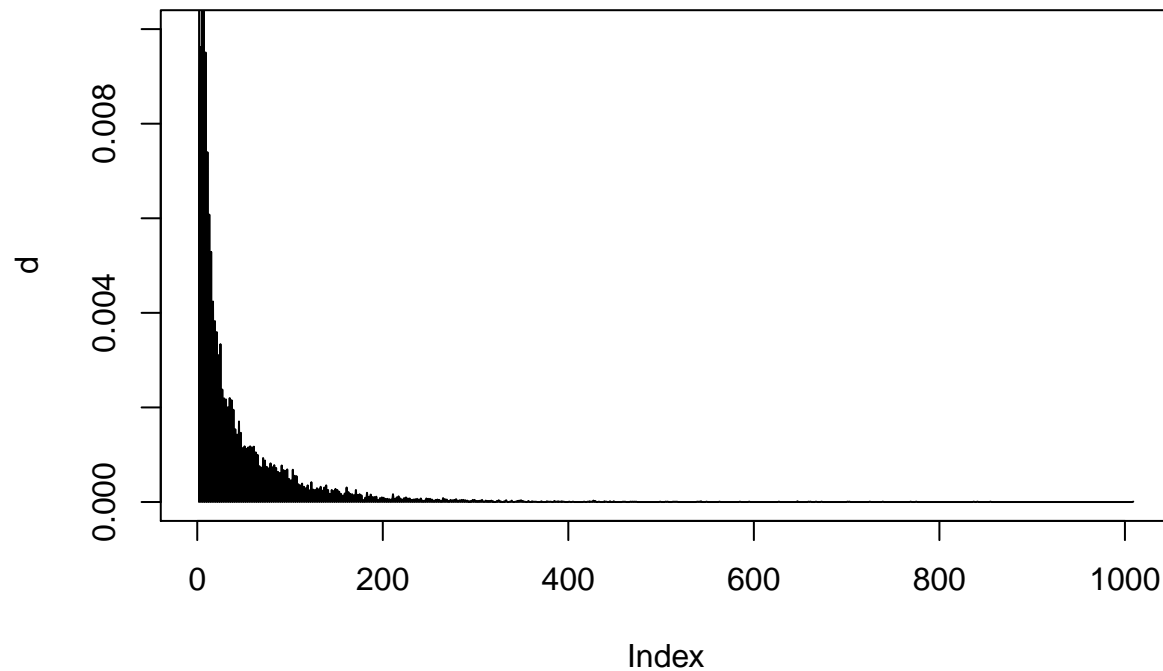
```
## [1] "HITS"
```

```
maxC(c4)
```

```
## [1] 53213 37479 124529 84122 127886 10976 90794 91586 95036 75947
```

```
#id 53213 appears 3 times as the most central one
```

```
g <- graph_from_edgelist(as.matrix(d), directed = F)  
d <- degree_distribution(g)  
plot(d, type='l', ylim=range(0,.01))
```



```
n <- 133279  
cl <- clusters(g)  
print(paste('      - number of components =', cl$no))
```

```
## [1] "      - number of components = 114797"
```

```
print('    - we see in the following table:')
```

```
## [1] "    - we see in the following table:"
```

```
t = table(cl$csizes)
print(t)
```

```
##
##      1      2      3      4      5      6      7      8      9     10
## 114508  140    84    36    14     3     3     3     1     2
##      12    18 17903
##      1      1      1
```

```
print('')
```

```
## [1] ""
```

```
print(paste('    that the largest component captures', round(17903/n*100, 0), 'percent of the nodes.'))
```

```
## [1] "    that the largest component captures 13 percent of the nodes."
```

```
print('But this cluster is still much bigger than the second one.')
```

```
## [1] "But this cluster is still much bigger than the second one."
```